

Eero Mäkiranta

# VENYVÄN ELEKTRONIIKAN KIERTO- JA VENYTTESTILAITE

Informaatioteknologian ja viestinnän tiedekunta  
Kandidaatin tutkinto  
Joulukuu 2019

# TIIVISTELMÄ

Eero Mäkiranta: Venyvän elektronikan kierto- ja venytystestilaite  
Tampereen yliopisto  
Tieto- ja sähkötekniikan TkK-tutkinto-ohjelma, Sähkötekniikka  
Kandidaatintyö  
Joulukuu 2019

---

Venyvä elektronikka on kehittyvä ja paljon tutkittu tapa tehdä elektronisia ratkaisuja. Jotta varmasti toimivaa ja kestäväää venyvää elektronikkaa voidaan valmistaa, täytyy sitä testata ensin huolellisesti. Tämän työn tarkoituksena on suunnitella ja rakentaa kierto- ja venytystestilaite juuri tähän käyttötarkoitukseen. Koska materiaalin venyminen ja kiertyminen on venyvän elektronikan sovelluskohteissa yleistä, datankeruu juuri tällä testausmenetelmällä on hyvin perusteltua ja kerättyä dataa voidaan käyttää hyväksi valmistusmenetelmien parantamiseksi.

Tässä työssä rakennettiin pelkästään laitteen sähköinen osuus, joka sisältää ohjausjärjestelmän sekä moottorit. Työn lopputuloksena oli onnistunut järjestelmä, joka täytti kaikki laitteen toiminnalliset vaatimukset. Koska laite ei ole vielä valmis, täytyy sitä vielä jatkokehittää. Laitteeseen tulee rakentaa vielä testausjärjestelmä, joka mittaa näytteen sähköisiä ominaisuuksia. Myöskin laitteen mekaaninen kokoonpano täytyy toteuttaa, kunhan kaikki osa-alueet ovat valmistuneet.

Avainsanat: Venyvä elektronikka, kierto, venytys, testilaite, lineaarimoottori, askelmoottori, Arduino, ohjaus

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

# ALKUSANAT

Tämä työ on toteutettu yhteistyössä sähkötekniikan yksikön kanssa ja tässä työssä rakennettava laite päättyy sinne lopulta yhdeksi monista testilaitteista. Idea työn toteuttamiseen tuli kandidaatintyöseminaarin aihepankista ja otettuani yhteyttä ohjaavaan opettajaan, työn aiheeksi muodostui testilaitteen rakentaminen. Erityiskiitos Vänni Panulalle, joka oman työnsä ja opiskelunsa ohessa auttoi kandidaatintyön kanssa erittäin paljon.

Tampereella, 12.12.2019

Eero Mäkiranta

# SISÄLLYSLUETTELO

1. JOHDANTO .....	6
2. VENYVÄN ELEKTRONIIKAN TESTAAMINEN .....	7
3. TESTILAITTE .....	11
3.1 Laitteen suunnittelu .....	11
3.2 Laitteen rakentaminen .....	14
3.2.1 Kytkenät .....	15
3.2.2 Ohjelmointi .....	21
3.2.3 Piirilevyn valmistaminen .....	24
3.2.4 Laitteen toiminta .....	27
4. JOHTOPÄÄTÖKSET .....	28
LÄHTEET .....	29
LIITE A: OHJELMAKOODI .....	30

## KUVALUETTELO

<i>Kuva 1. Esimerkki työssä käytettävästä kalvosta.</i>	7
<i>Kuva 2. Näytteen venytystestaus [3].</i>	8
<i>Kuva 3. Rullaustesti [3].</i>	9
<i>Kuva 4. Rullaus akselin ympärille ja levyjen välissä [3].</i>	9
<i>Kuva 5. Alustava suunnitelma kierto- ja venytystestilaitteesta.</i>	11
<i>Kuva 6. Lohkokaavio laitteen sähköisestä toiminnasta.</i>	12
<i>Kuva 7. Lohkokaavio Arduinon ohjelmoinnista.</i>	13
<i>Kuva 8. Moottoriohjain.</i>	15
<i>Kuva 9. Moottoriohjaimen kytkeminen [8].</i>	16
<i>Kuva 10. Moottoriohjaimen yläosan liittimet [8].</i>	16
<i>Kuva 11. Moottoriohjaimen alaosan liittimet [8].</i>	17
<i>Kuva 12. Askelmoottorin kytkennät.</i>	17
<i>Kuva 13. Painonapin kytkentä.</i>	18
<i>Kuva 14. Kytkentä ilman lineaarimoottoria.</i>	19
<i>Kuva 15. Lineaarimoottorin ohjaimen sisäänmenoliitin.</i>	19
<i>Kuva 16. Releen ja transistorin kytkentä.</i>	20
<i>Kuva 17. Lopullinen prototyyppikytkentä.</i>	21
<i>Kuva 18. Moottoreiden liikuttaminen.</i>	23
<i>Kuva 19. End_cycle-funktio.</i>	24
<i>Kuva 20. Kytkennän piirikaavio.</i>	25
<i>Kuva 21. Piirilevylayout.</i>	25
<i>Kuva 22. Valmis piirilevy.</i>	26
<i>Kuva 23. Testausjärjestely.</i>	27

## LYHENTEET JA MERKINNÄT

GND	Maapotentiaali
HIGH	Arduinon ohjelmointimerkintä, joka tarkoittaa 5 V:n jännitettä
int	Integer, kokonaisluku
LOW	Arduinon ohjelmointimerkintä, jolla tarkoittaa 0 V:n jännitettä
mm	Millimetri
N	Newton
VAC	Vaihtojännite
VDC	Tasajännite

# 1. JOHDANTO

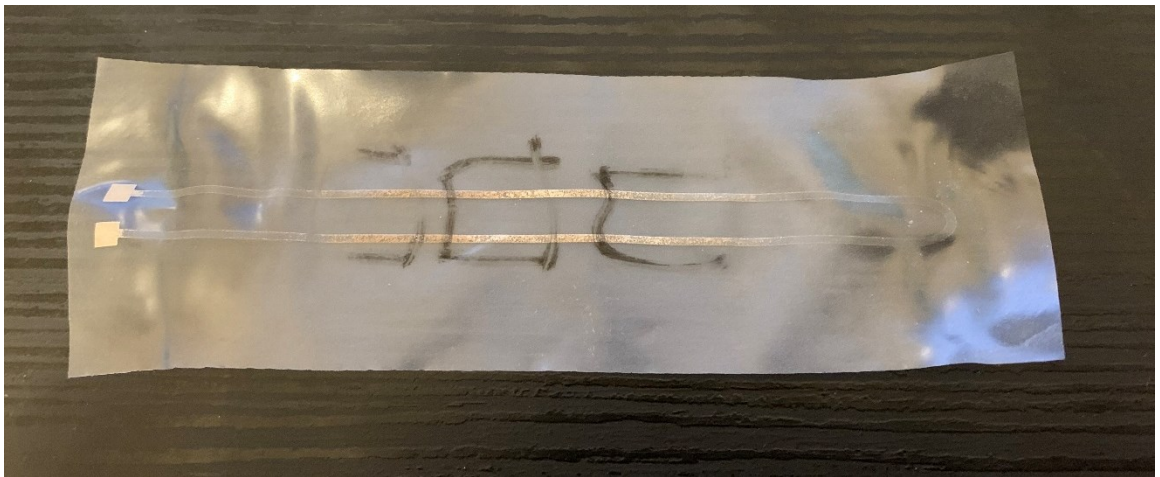
Painettava elektroniikka on jatkuvasti kehittyvä ja tutkittava tapa tehdä elektronisia toteutuksia. Painettava elektroniikka on nimensä mukaisesti elektronisten toteutusten painamista tai tulostamista erilaisten materiaalien päälle. Tällä tavoin voidaan saavuttaa ominaisuuksia, joita esimerkiksi perinteisellä piirilevyllä ei ole. Näitä ominaisuuksia ovat esimerkiksi ohuus, joustavuus, puettavuus sekä keveys. Painettavaa elektroniikkaa voidaan käyttää esimerkiksi lääketieteessä ihmisten elintoimintojen mittaamisessa. Lisäksi painettavan elektroniikan avulla on mahdollista sulauttaa vaatteisiin erilaisia antureita, jotka mittaavat ihmisten elintoimintoja. [1]

Koska painettava elektroniikka kehittyy jatkuvasti, tulee sitä myös testata huolellisesti. Tämä työ käsittelee erityisesti venyvää elektroniikkaa ja työn tarkoituksena on valmistaa kierto- ja venyttestilaitteisto venyvän elektroniikan testaukseen. Laite tulee venyttämään sekä kiertämään venyvää termoplastista polyuretaanipalaa, jonka pinnalle on painettu elektroniikkaa. Testilaitteen tarkoituksena on tutkia näytteen pinnalla olevan elektroniikan kestävyyttä, kun se altistetaan jatkuville venymissykleille. Laitteella pystytään siis määrittämään se syklimäärä, jonka jälkeen näytteen pinnalla olevan elektronisen piirin resistanssi muuttuu merkittävästi, mikä tarkoittaa painetun elektroniikan vaurioitumista. Tässä työssä ei kuitenkaan tulla käsittelemään kestävyuden mittausta, vaan työ keskittyy pelkästään testauslaitteistoon ja sen rakentamiseen.

Työ koostuu neljästä luvusta, joista toisessa luvussa käsitellään painettavan elektroniikan teoreettista taustaa, sekä syitä ja erilaisia menetelmiä elektroniikan testaukselle. Kolmannessa luvussa käsitellään testilaitteiston suunnittelu- sekä valmistusprosessia ja esitellään valmiin laitteen toiminnallisuutta. Neljännessä luvussa esitellään johtopäätökset ja pohditaan, onko halutut toiminnallisuudet saavutettu. Viimeisessä luvussa käsitellään myös laitteen jatkokehitystä ja sen parantamista.

## 2. VENYVÄN ELEKTRONIIKAN TESTAAMINEN

Venyvällä elektroniikalla tarkoitetaan piirejä, joita voidaan taivuttaa sekä venyttää ilman, että niiden rakenne tuhoutuu. Venyvä elektroniikka sisältää kaikki tavalliset niin sanotun jäykän elektroniikan ominaisuudet, mutta esimerkiksi perinteisestä piirilevystä poiketen, venyvä elektroniikka kestää erilaiset mekaaniset muodonmuutokset [2]. Venyvää elektroniikkaa valmistetaan painamalla tai tulostamalla halutut ominaisuudet omaavan materiaalin pinnalle sähköä johtavaa materiaalia. Tässä työssä valmistettava kierto- ja venyttestilaite tulee testaamaan kuvan 1 mukaista venyvää kalvoa, jonka pohjamateriaalina käytetään termoplastista polyuretaania.



**Kuva 1. Esimerkki työssä käytettävästä kalvosta.**

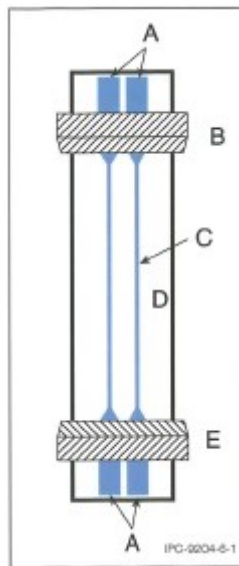
Työssä käytettäviä kalvoja on kahta eri tyyppiä. Kuvassa 1 on esitetty kalvo, joka sisältää pelkät johtimet. Toinen kalvotyyppi on hyvin edellä mainittua kalvoa vastaava, mutta siihen on myös lisätty johtimien väliin kontaktipinnat halutuille passiivikomponenteille.

Venyvän elektroniikan testaukselle ei ole olemassa varsinaisia standardeja, sillä ala on vielä kehitysvaiheessa. Aiheesta löytyy kuitenkin dokumentteja, jotka toimivat yleisenä ohjenuorana venyvän elektroniikan testaamiselle. Esimerkiksi IPC on julkaissut vuonna 2017 tällaisen dokumentin [3]. Dokumentissa käsitellään erilaisia yleisiä testausmenetelmiä, joita voidaan soveltaa painettavan elektroniikan testaamiseen.

Venyvän elektroniikan testaaminen voidaan jakaa viiteen kategoriaan: Venytys, taivutus, kierto, rullaus ja rypistäminen [3]. Käytettävä testauksen kategoria riippuu testattavan materiaalin käyttökohteista.



Venytystestien tarkoituksena on arvioida venyvän elektronikan toiminnallisuutta jatkuvan tai sykleissä toistettavan venymiskuormituksen vaikutuksen alaisuudessa. Venytystesteihin soveltuva näytteen kiinnitystapa on esitetty kuvassa 2.

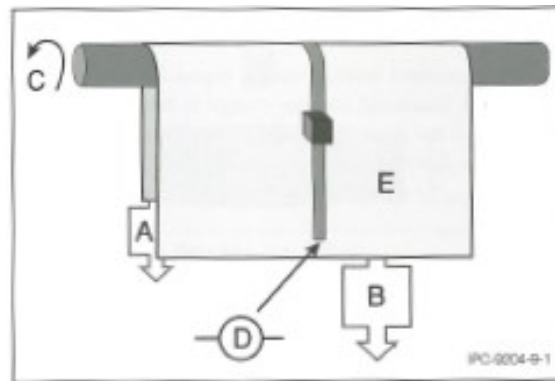


**Kuva 2. Näytteen venytystestaus [3].**

Venymisrajatestillä määritetään se prosentuaalinen määrä, jota näytekalvoa voidaan venyttää ennen kuin sen pinnalle painettu piiri pettää sähköisesti tai mekaanisesti. Pettäminen todetaan seuraamalla jatkuvasti piirin toiminnallisuutta tai resistanssia. Testi voidaan suorittaa venyttämällä näytekalvoa pelkästään toisesta päästä tai venyttämällä molempia päitä vastakkaisiin suuntiin. Venymissyklitestillä määritetään, montako venyttämisykliä näyte kestää ennen kuin sen toiminnallisuus tai sähköiset ominaisuudet pettävät. Näytettä venytetään tietty prosentuaalinen määrä ja halutun pitoajan jälkeen näyte palautetaan takaisin alkuperäiseen tilaansa, jonka jälkeen sykli toistetaan uudelleen. Venytyskestävyyttä voidaan tutkia myös testillä, jossa näytettä venytetään tietty prosentuaalinen määrä ja jätetään venytettyyn tilaan. Tämän jälkeen näytteen toiminnallisuutta ja resistanssia mitataan, minkä avulla voidaan määrittää, kauanko näyte kestää jatkuvaa venytystä. [3] Venymistestit voidaan myös yhdistää kiertotestien kanssa, kuten tässä työssä valmistettavan kierto- ja venytystestilaitteen tapauksessa. Yksi tapa on kiertää näytettä haluttu määrä, jonka jälkeen toimitaan samalla tavalla kuin venymissyklitestissä. Toinen tapa vastaa edellä mainittua, mutta myös kierto palautetaan lähtötilaansa syklin lopuksi. Näytettä siis kierretään haluttu määrä, jonka jälkeen sitä venytetään venymissyklitestin mukaisesti. Pitoajan jälkeen sekä venytys- että kiertokuormitus puretaan ja näyte palaa takaisin lähtötilaansa. [3] Tässä työssä rakennettava testilaitte noudattelee perusperiaatteeltaan viimeksi mainittua testitapaa.

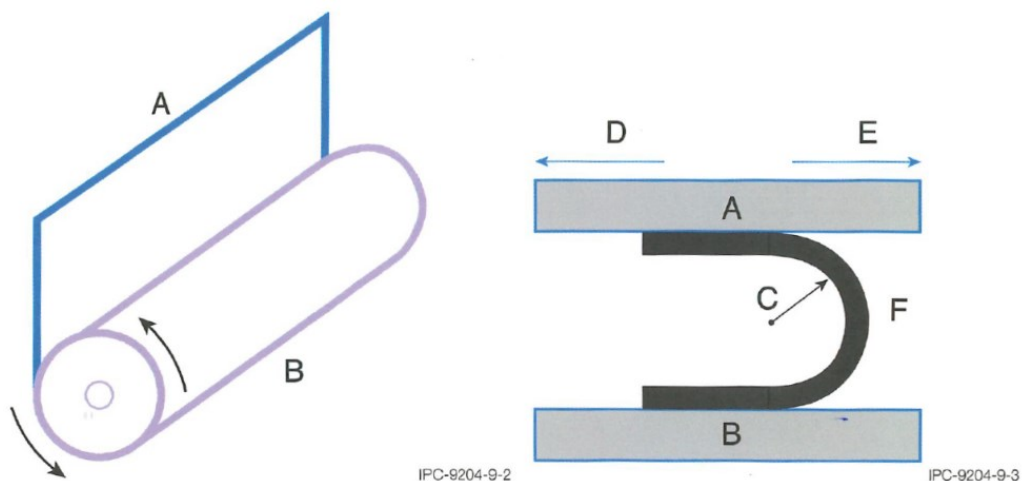
Taivutustestien tarkoituksena on määrittää venyvän elektronikan kestävyys erilaisissa taivutusolosuhteissa. Taivutuskestävyyttä voidaan testata esimerkiksi seuraavasti: Eri taivutussäteillä suoritettavilla taivutustesteillä, eri taivutuskulmalla suoritettavilla taivutustesteillä, taivutusta ja venytystä vuorotellen aiheuttavilla testeillä sekä taittamiskestävyttä mittaavilla testeillä. [3]

Rullaustestit ovat myös yleinen tapa testata venyvän elektronikan ja sen pinnalla olevien komponenttien kestävyttä. Rullauskestävyyttä voidaan tutkia esimerkiksi liikuttamalla näytettä pitkittäissuunnassa rullan yli edestakaisella liikkeellä kuvan 3 mukaisesti.



**Kuva 3. Rullaustesti [3].**

Toinen tapa testata rullauskestävyyttä on rullata näyte kokonaan akselin ympärille, jonka jälkeen näyte jälleen suoritetaan. Rullauskestävyyttä voidaan myös testata asettamalla näyte kahden levyn väliin siten, että levyjen välissä näyte on taittuneena halutulle taivutussäteelle. Tämän jälkeen toista levyä liikutetaan edestakaisin, jolloin näyte kokee rullausrasitusta. [3] Nämä tavat on esitetty kuvassa 4.



**Kuva 4. Rullaus akselin ympärille ja levyjen välissä [3].**

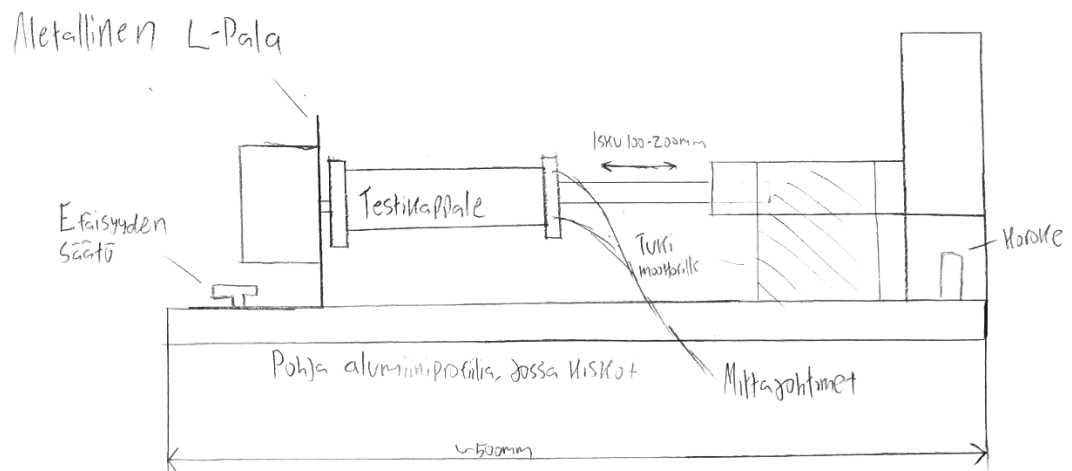
Rullaustestejä suoritetaan useita syklejä, kunnes näytteen toiminnallisuudessa havaitaan poikkeama. Tällainen poikkeama saattaa olla esimerkiksi resistanssin muutos, joka johtuu johtimen tai näytteessä kiinni olevan komponentin kiinnityksen pettämisestä.

### 3. TESTILAITTE

Testilaitteen suunnittelun ja rakentamisen tueksi ei ollut saatavilla mallia, jota olisi voinut käyttää tukena, sillä vastaavista laitteista ei ollut saatavilla referenssimateriaalia. Laitte täytyi siis suunnitella ja rakentaa alusta alkaen itse käyttäen hyväksi ainoastaan itse määritettyjä toiminnallisia vaatimuksia.

#### 3.1 Laitteen suunnittelu

Kääntö- ja venytystestilaitteen suunnittelu aloitettiin määrittelemällä laitteen toiminnalliset vaatimukset. Nämä vaatimukset määriteltiin yhdessä professori Matti Mäntysalon kanssa. Yleisin testattava kappale on pituudeltaan noin 120 mm ja leveydeltään noin 40 mm. Koska testattavan materiaalin tulee kestää myös suurta mekaanista rasitusta ilman, että sen pinnalle painettujen johdinten sähköinen toiminta muuttuu, tulee laitteen kyetä venyttämään testikappaletta vähintään 120 mm. Tällöin testilaitte pystyy rasittamaan kappaletta tarpeeksi. Myös kierto on erittäin tärkeä osa testausta, sillä myös se aiheuttaa kappaleelle erittäin suurta rasitusta. Kierron minimivaatimukseksi määriteltiin 360 astetta, kuitenkin siten, että se on säädeltävissä. Laitteen tulee olla toimintavarma, sillä yhtä testattavaa kappaletta tulee testata jopa tuhannen syklin verran. Yksi sykli koostuu venytyksestä, kierrosta sekä palautuksesta alkuperäiseen tilaan. Kuvassa 5 on esitetty määriteltyjen toiminnallisten vaatimusten pohjalta piirretty alustava kuva laitteesta.



**Kuva 5. Alustava suunnitelma kierto- ja venytystestilaitteesta.**

Suunnitelman mukaan venytykseen käytetään lineaarimoottoria. Lineaarimoottori on toimilaite, joka muuttaa sähkömoottorin kiertävän liikkeen pituussuuntaiseksi liikkeeksi. Li-

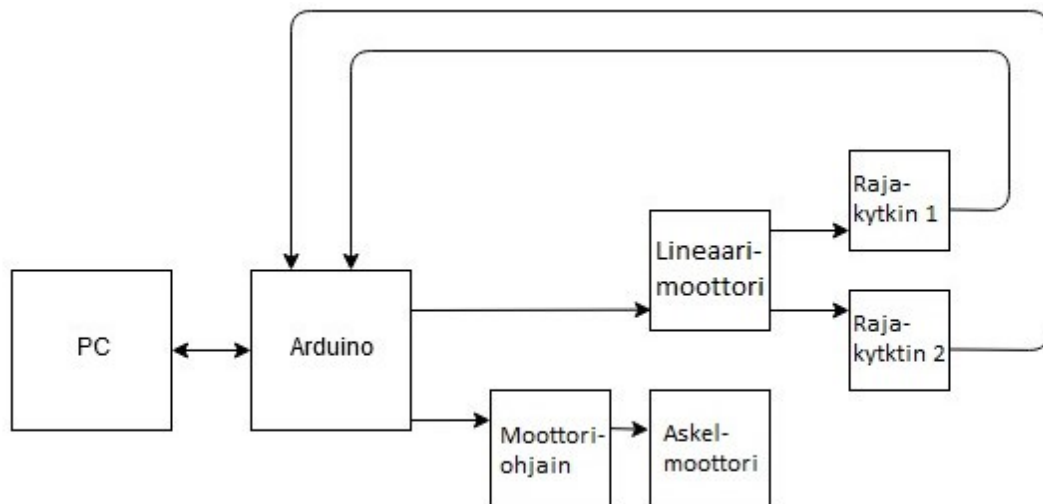
neaarimoottori koostuu perinteisestä pyörivästä sähkömoottorista, joka liikuttaa lineaarimoottorin sisällä olevaa karaa. Sähkömoottorin pyörimissuunta määrittää sen, kumpaan suuntaan kara liikkuu eli työntyykö lineaarimoottorin varsi ulospäin vai vetäytyykö se sisään. [4]

Kiertoon käytetään askelmoottoria, jolloin testikappaleen kierron määrää on helppo säädellä. Askelmoottori on sähkömoottori, jota pystytään pyörittämään tarkoin määritellyissä askeleissa toisin kuin tavallista sähkömoottoria, jonka pyöriminen on jatkuvaa. Askelmoottori pyörähtää askeleen kerrallaan, ja yhden askeleen suuruus määräytyy askelmoottorin rakenteesta. [5] Askelmoottoria pystyy kuitenkin pyörittämään tavallisen sähkömoottorin tavoin ja askeleita ei huomaa sen pyöriessä.

Lineaarimoottorin liikesuunnan määrittämiseen käytetään mikrokytkimiä, jotka toimivat tässä työssä käsiteltävässä laitteessa rajakytkiminä. Kun lineaarimoottori on liikkunut halutun määrän, sen varsi aktivoi laitteessa olevat rajakytkimet ja laitetta ohjaava ohjelma kääntää lineaarimoottorin liikesuunnan.

Laitteen runkona käytetään alumiiniprofiilia, jonka leveys on vähintään 135 mm ja pituus 500 mm. Alumiiniprofiilin kiskoihin on helppo toteuttaa kiinnitykset moottoreille käyttäen hyväksi metallisia L-paloja.

Laitteen sähköisestä toiminnasta piirrettiin lohkokaavio, jota käytettiin hyväksi laitteen suunnittelussa. Lohkokaavio on esitetty kuvassa 6.



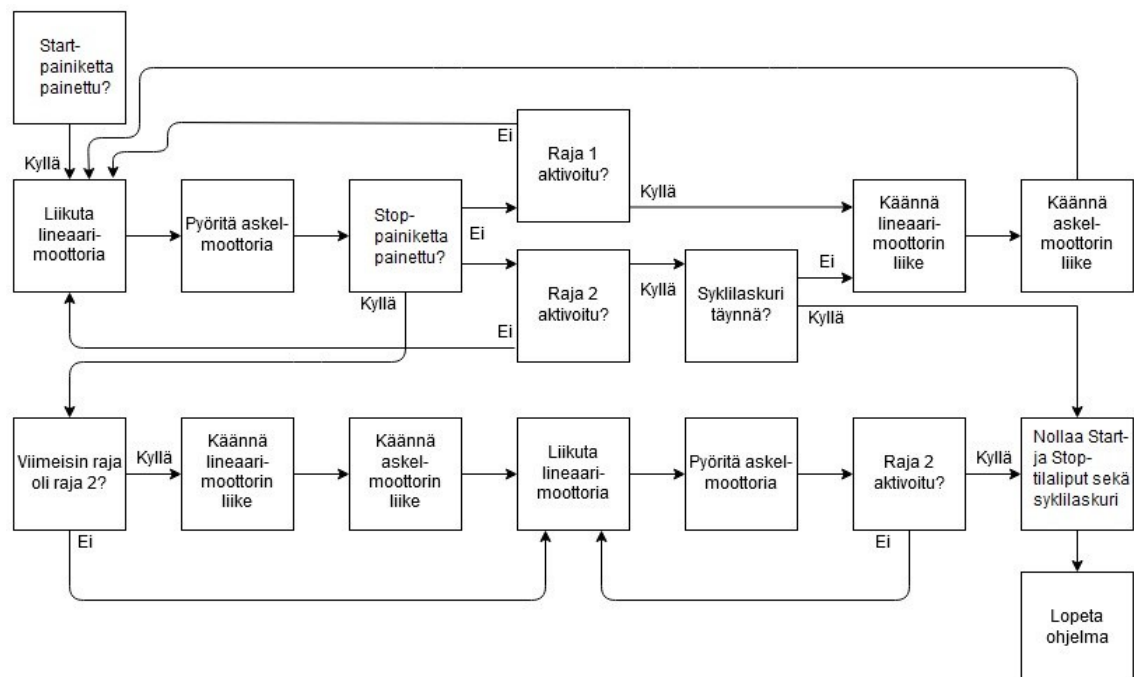
**Kuva 6. Lohkokaavio laitteen sähköisestä toiminnasta.**

Laitteen ohjaus tapahtuu suunnitelman mukaan Arduino-kehitysalustalla, joka ohjaa lineaarimoottoria sekä askelmoottorin ohjainta. Arduinon ohjelmoimiseen käytetään C- ja C++ -kieliin pohjautuvaa Arduino-ohjelmointikieltä [6]. Lineaarimoottorin liikettä ohjataan

kahdella mikrokytkimellä, jotka aktivoituvat, kun lineaarimoottori saavuttaa halutun kohdan. Mikrokytkimien aktivoiduttua lineaarimoottori vaihtaa suuntaa. Mikrokytkimien antama signaali tulee vaikuttamaan myös askelmoottorin kiertosuuntaan.

Laitteiston tehonsyöttö toteutetaan laboratoriovirtalähteellä, jolla saadaan säädettyä laitteistolle tarvittavat virta- ja jännitearvot. Lineaarimoottorin tehonsyöttö toteutetaan moottorin omalla ohjaimella, joka toimii 230 VAC:n jännitteellä ja joka syöttää lineaarimoottorille 24 VDC:n jännitettä.

Laitteen ohjauksessa käytettävän Arduino-kehitysalustan ohjelmointia havainnoitiin lohkokaaviolla. Lohkokaavio on esitetty kuvassa 7.



**Kuva 7. Lohkokaavio Arduinon ohjelmoinnista.**

Ohjelman suoritus aloitetaan painamalla start-painiketta, joka on laitteessa oleva fyysinen painike. Start-painikkeen painaminen asettaa tilalipun, joka estää start-painikkeen toiminnallisuuden toistumisen, jos sitä painetaan kesken ohjelman suorituksen. Ohjelma pitää jatkuvasti kirjaa siitä, kuinka monta sykliä testausta on suoritettu. Tämä syklimäärä on ennalta asetettu ohjelmaan. Start-painikkeen painamisen jälkeen lineaarimoottori alkaa venyttää testikappaletta ja askelmoottori aloittaa kappaleen kiertämisen. Tätä suoritetaan niin kauan, kunnes toinen rajakytkimestä aktivoituu. Rajakytkimen aktivoitumisen jälkeen ohjelma kääntää sekä lineaarimoottorin liikesuunnan että askelmoottorin kiertosuunnan. Sykliä määrää tarkastellaan sen rajakytkimen kohdalla, joka määrää laitteen perustilaa eli sitä tilaa, jossa testikappale on löysänä tai suorana. Jos ohjelma havaitsee,

että syklimäärä on tullut täyteen, nollaa se sekä start- että stop-tilaliput ja lopettaa ohjelman.

Ohjelmassa on myös keskeytyspainike (stop), joka keskeyttää ohjelman, mikäli sitä painetaan. Tätä keskeytyspainiketta tarkastellaan ennen rajojen aktivoitumisen tarkastelua. Mikäli stop-painiketta painetaan, ohjelma keskeyttää pääohjelman suorittamisen. Tämän jälkeen ohjelma tarkastaa viimeisimmän aktivoituneen rajakytkimen. Mikäli viimeisin rajakytkin on ollut perustilan rajakytkin, ohjelma kääntää sekä lineaarimoottorin liikesuunnan että askelmoottorin kiertosuunnan ja lähtee liikuttamaan lineaarimoottoria kohti perustilan rajakytkintä. Kun perustilan rajakytkin on aktivoitunut, ohjelma nollaa kaikki tilaliput ja lopettaa ohjelman. Mikäli keskeytyspainikkeen painamisen hetkellä viimeisin aktivoitunut rajakytkin on ollut venyneen tilan rajakytkin, toisin sanoen ohjelma on palaa-  
massa perustilaan, ohjelma odottaa, että perustilan rajakytkin aktivoituu, minkä jälkeen se nollaa tilaliput ja lopettaa ohjelman.

Laitteessa käytettävät komponentit valittiin edellä läpi käytyjen suunnitelmien ja vaatimusten perusteella. Suunnitelmia ei voitu kuitenkaan noudattaa täysin, johtuen komponenttien saatavuudesta. Varsinkin lineaarimoottori aiheutti suunnitelmien muutoksia, sillä täysin suunnitelmien mukaista lineaarimoottoria ei löytynyt. Lineaarimoottoriksi valittiin RS PRO FD60 -moottori, joka kykenee 300 mm liikkeeseen ja 3000 N:n vetävään voimaan. Kyseinen lineaarimoottori toimii 24 VDC jännitteellä. Lineaarimoottorin ohjaukseen ja tehonsyöttöön valittiin RS PRO T-230-24-1 -ohjain. [7] Askelmoottoriksi valittiin SparkFun Electronics 1568-1105-ND 12 V -bipolaarinen askelmoottori. Askelmoottorin ohjaimeksi valittiin SparkFun Electronics 1568-1108-ND ja mikrokytkimiksi valittiin Omron Electronics Z4701-ND -mikrokytkimet. Testilaitetta ohjataan Arduino Uno -kehitysalustalla. Lopullinen ohjauspiiri rakennetaan itse valmistetulle piirilevyille. Edellä mainittujen komponenttien lisäksi tarvitaan kaksi relettä sekä kaksi transistoria lineaarimoottorin ohjaimen ohjaamiseen.

### 3.2 Laitteen rakentaminen

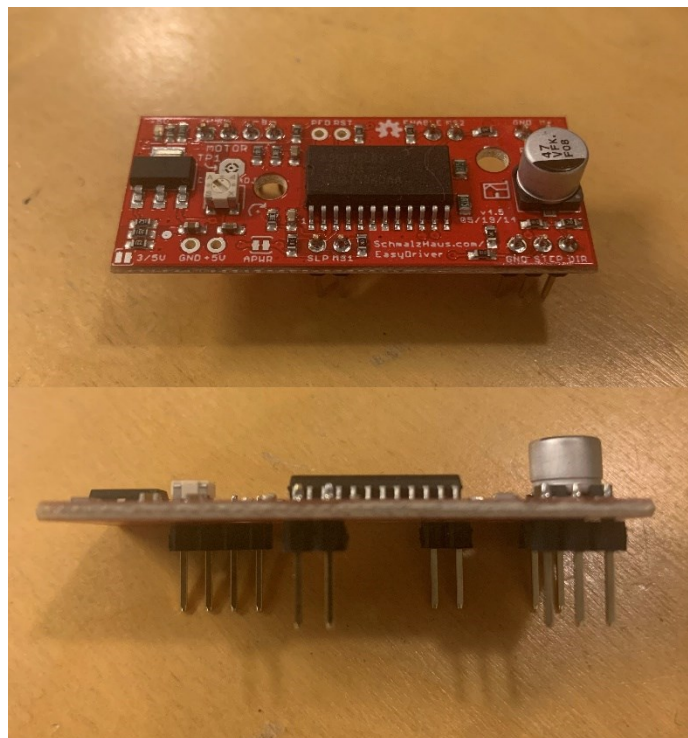
Laitteen rakentaminen aloitettiin tutkimalla laitetta varten hankittuja komponentteja sekä niiden datalehtiä. Ensimmäinen vaihe rakentamisessa oli saada askelmoottori toimimaan halutulla tavalla. Askelmoottori tuli siis kytkeä moottoriohjaimeen, joka kytkettiin taas Arduinoon. Tähän kytkentään löytyi erittäin hyvät ohjeistukset moottoriohjaimen datalehdeltä. Tämän jälkeen Arduino ohjelmoitiin ohjaamaan moottoriohjainta halutulla tavalla, joka taas suorittaa askelmoottorin ohjaamisen.

Kun askelmoottorin toiminnallisuus saatiin toteutettua, järjestelmään lisättiin tarvittavat painikkeet sekä rajakytkimet. Prototyypivaiheessa rajakytkimiä havainnollistettiin tavallisilla normal open -painikkeilla. Normal open tarkoittaa siis sitä, että piiri on avoin, kunnes painiketta painetaan, jolloin piiri sulkeutuu. Lopulliset mikrokytkimet, joita käytetään rajakytkiminä, toimivat samalla periaatteella. Nämä painikkeet liitettiin Arduinoon ja Arduinon ohjelmaan lisättiin toiminnallisuus painikkeille.

Viimeinen osa laitteen toiminnallisuuden toteuttamista oli lineaarimoottorin kytkeminen järjestelmään. Lineaarimoottorin kytkeminen jätettiin viimeiseksi, sillä sen kanssa oli ongelmia. Lineaarimoottorin datalehti oli hyvin vajavainen, joten lisätietoa lineaarimoottorin kytkemisestä ja käyttämisestä täytyi kysyä suoraan toimittajalta, jonka vastausajat olivat erittäin pitkiä. Lopulta lineaarimoottorin ohjaus päätettiin toteuttaa valmiilla ohjaimella, joka tilattiin lineaarimoottoria varten.

### 3.2.1 Kytkennät

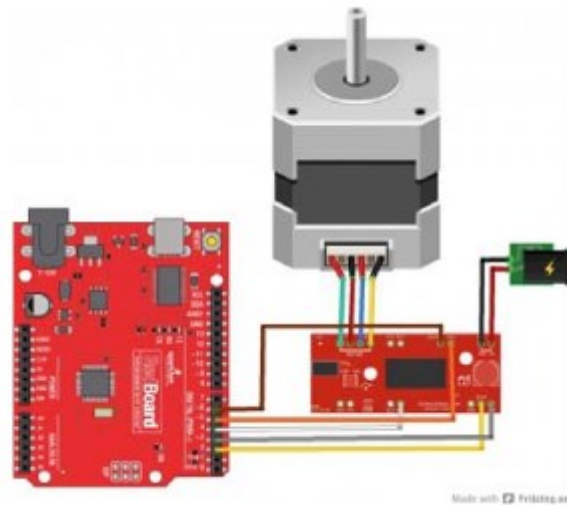
Aluksi laitteesta rakennettiin prototyyppi, joten kaikki kytkennät tehtiin koekytkentälevylle. Kytkennät aloitettiin askelmoottorin moottoriohjaimen liittamisestä kytkentälevylle. Askelmoottorin läpivienteihin juotettiin piikkirimaa kuvan 8 mukaisesti, jolloin moottoriohjain saatiin liitettyä kytkentälevyyn.



**Kuva 8. Moottoriohjain.**



Tämän jälkeen itse askelmoottori yhdistettiin moottoriohjaimeen. Moottorin kytkemisessä käytettiin apuna moottoriohjaimen datalehteä, josta kävi ilmi, miten moottori tulisi kytkeä kytkentälevylle. Askelmoottorista tulevat neljä johdinta kytkettiin kuvan 9 mukaisesti.



**Kuva 9. Moottoriohjaimen kytkeminen [8].**

Moottorissa on kaksi käämiä, jotka saavat aikaan moottorin liikkeen, kun niihin syötetään virtaa. Molemmille käämeille menee kaksi johdinta, jotka on värikoodattu. Moottoriohjaimessa käämeille lähtevät ulostulot on merkattu merkinnöin A ja B, joista kummallakin on plus- ja miinus -liittimet. Kyseiset liittimet on esitetty kuvassa 10.



**Kuva 10. Moottoriohjaimen yläosan liittimet [8].**

Käämeille menevät johtimet on määritetty siten, että vihreä sekä punainen johdin menevät toiselle käämille ja sininen sekä keltainen johdin menevät toiselle käämille. [8] Askelmoottorin johtimet kytkettiin koekytkentälevylle siten, että vihreä johdin kytkettiin moottoriohjaimen A+ -liitäntään ja punainen johdin liitettiin A- -liitäntään. Samanlainen kytkentä toistettiin toisen käämin kohdalla, eli sininen johdin kytkettiin B+ -liitäntään ja keltainen johdin kytkettiin B- -liitäntään. Koska käämeillä ei ole polariteettia, kytkettäessä ei tarvinnut huolehtia siitä, kuinka päin samalta käämiltä tulevat johtimet kytketään ohjaimelle [8].

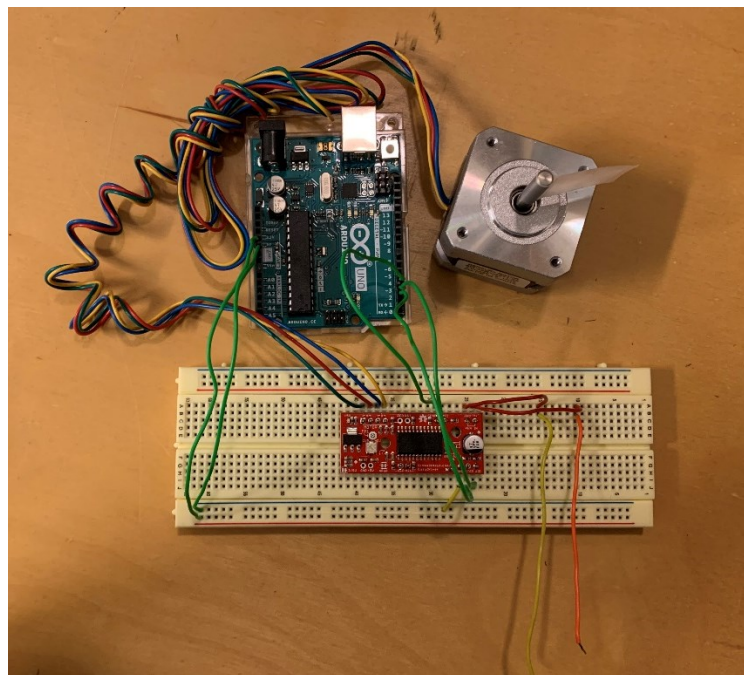
Moottorin kytkemisen jälkeen moottoriohjain kytkettiin Arduinin ulostuloihin. Tässä työssä askelmoottori tarvitsi vain yksinkertaisen ohjauksen, joten kuvasta kuusi poiketen Arduinoon kytkettiin pelkästään moottoriohjaimen sisäänmenot STEP, DIR sekä ENABLE. STEP-sisäänmeno saa sisäänsä pulsseina joko 0 V tai 5 V (LOW tai HIGH), joka aiheuttaa moottorin pyörimisen. Pulssien tiheys vaikuttaa askelmoottorin pyörimis-

nopeuteen. DIR-sisäänmeno määrittää askelmoottorin pyörimissuunnan. Pyörimissuunta vaihtuu siis sen mukaan, näkyykö sisäänmenossa 0 V:n vai 5 V:n jännite. ENABLE-sisäänmeno määrittää ajaako moottoriohjain moottoria. ENABLE perustuu FET-transistoreihin, joita käytetään ohjaimessa kytkiminä. Jos sisäänmenossa näkyy 5 V:n jännite, transistorit eivät johda ja ohjain ei aja moottoria. 0 V:n jännitteellä taas transistorit alkavat johtaa ja ohjain kykenee ajamaan moottoria. [8] STEP- sekä DIR-liittimet on esitetty kuvassa 11, ja ENABLE-liitin on esitetty kuvassa 10.



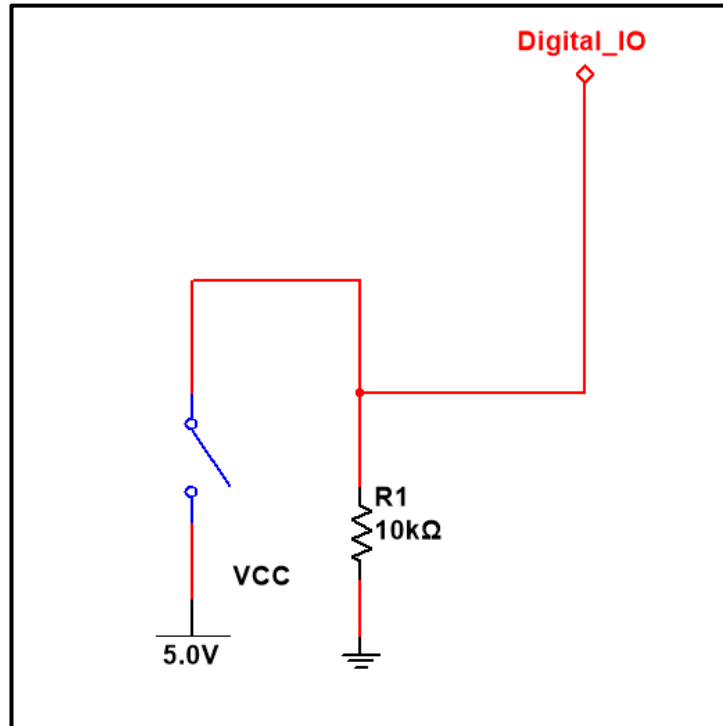
**Kuva 11. Moottoriohjaimen alaosan liittimet [8].**

Nämä ohjaimen sisäänmenot kytkettiin Arduinon digital input/output liittämiin siten, että STEP kytkettiin Arduinon liittimeen 2, DIR liittimeen 3 ja ENABLE liittimeen 4. Mikäli Arduinon digital input/output liitin on asetettu ulostuloksi, se kirjoittaa liittimelle joko HIGH-tilan eli 5 V tai LOW -tilan eli 0 V. Input/output -määrittäminen sekä HIGH- tai LOW-tilan määrittäminen tehdään Arduinoa ohjelmoidessa. Lisäksi Arduinon oma 5 V:n käyttöjännitesyöttö kytkettiin hyppylangalla kytkentälevyn käyttöjännitekiskoon ja Arduinon omasta maaliitimestä (GND) kytkettiin hyppylanka kytkentälevyn maakiskoon. Lopuksi moottoriohjaimen GND -liitin kytkettiin koekytkentälevyn maakiskoon ja jännitelähde liitettiin kuvassa seitsemän näkyvään moottoriohjaimen M+ -sekä GND-liittimeen. Askelmoottorin sekä moottoriohjaimen kytkennät on esitetty kuvassa 12.



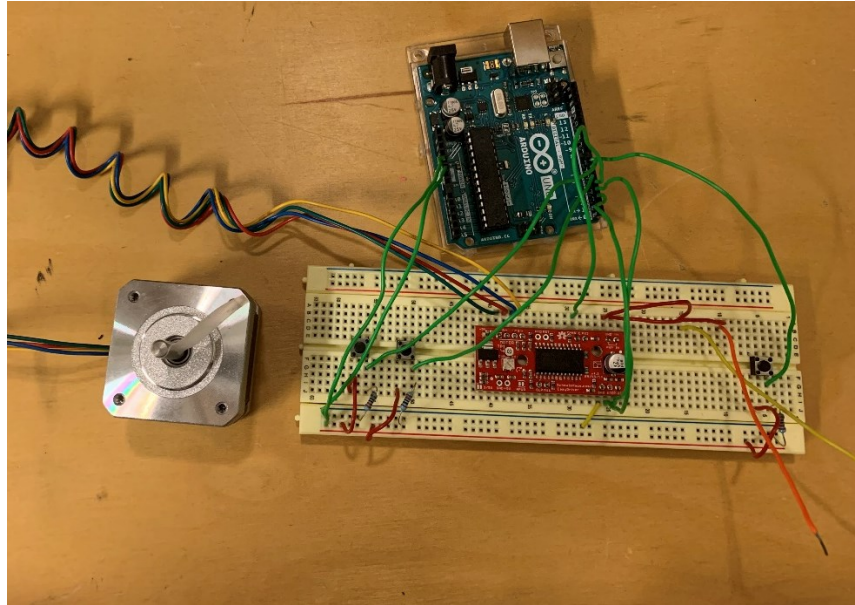
**Kuva 12. Askelmoottorin kytkennät.**

Kun askelmoottorin sekä moottoriohjaimen toiminta saatiin testattua, piiriin lisättiin kolme painonappia, joista kaksi kuvasti rajakytkimiä ja yksi toimi aloitus/keskeytys -painikkeena. Painonapit kytkettiin koekytkenälevyyn ja Arduinon digital input/output -liittimiin kuvan 13 mukaisesti. Painonapit kytkettiin Arduinon liittimiin viisi, kuusi ja seitsemän.



**Kuva 13. Painonapin kytkentä.**

Painonappi kytkettiin siis Arduinon syöttämän 5 V:n jännitteen ja maan välille. Heti painonapin jälkeen kytkettiin johdin, joka kytkeytyy Arduinon digital input/output liittimeen, jota tässä tapauksessa käytetään sisäänmenona. Piiriin lisättiin myös vastus estämään suora oikosulku käyttöjännitteen ja maan välille. Kun painonappia painetaan, Arduino tunnistaa 5 V:n jännitteen (HIGH), jota voidaan käyttää ohjelmallisesti järjestelmän ohjaamiseen. Painonappien sekä moottoriohjaimen kytkentä on esitetty kuvassa 14.



**Kuva 14. Kytkenä ilman lineaarimoottoria.**

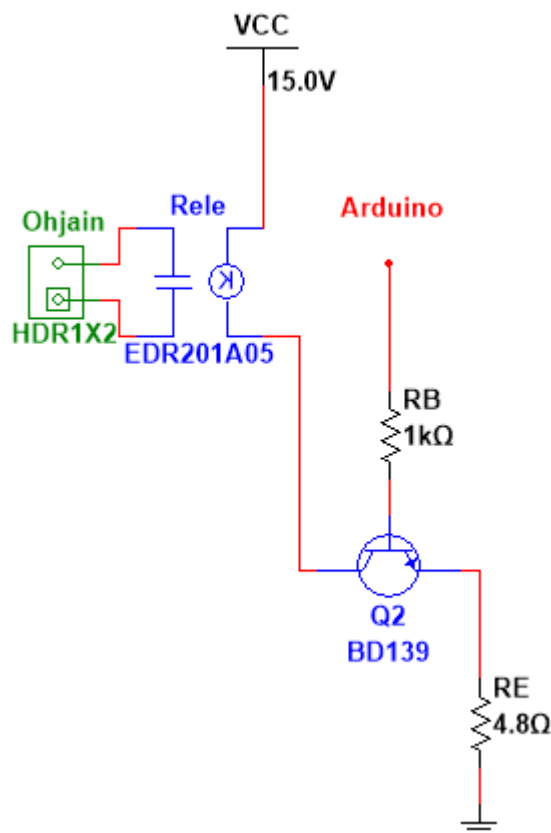
Lopuksi toteutettiin lineaarimoottorin kytkennät. Koska lineaarimoottorista ja siihen kuuluvasta ohjaimesta ei löytynyt datalehteä, täytyi niiden toiminnallisuus selvittää yleismitarin avulla. Aluksi selvitettiin, millaisella signaalilla lineaarimoottorin ohjainta ohjataan. Alkutietona oli, että ohjaimelle voidaan antaa käskyjä erikseen ostettavalla säätimellä, joka sisältää kaksi painiketta. Tämän vuoksi voitiin tehdä olettamus, että ohjaimen sisäänmenoliittimessä on kolme pinniä, joista yksi on jännitteinen ja loput kaksi toimivat nieluina. Se, kumpaan nieluun toimivaan pinniin jännite kytkeytyy, määrittää lineaarimoottorin liikesuunnan. Lineaarimoottorin ohjain, ja sisäänmenoliitin on esitetty kuvassa 15.



**Kuva 15. Lineaarimoottorin ohjaimen sisäänmenoliitin.**

Yleismittarilla havaittiin, että liittimen jännitteinen pinni on kuvassa 15 toinen vasemmalta. Tämän pinnin jännitearvoksi mitattiin noin 38 V. Yleismittarilla pystyttiin myös määrittämään ne kaksi pinniä, jotka toimivat nieluina. Määrittäminen suoritettiin laittamalla yleismittarin positiivinen johdin jännitteiseen pinniin ja etsimällä negatiivisella johtimella pinnit, joihin kytkeytyessään yleismittari näytti 38 V jännitearvoa. Tämän jälkeen aikaisemmin tehty oletamus ohjaimen toiminnasta todistettiin oikeaksi oikosulkemalla jännitteinen pinni sekä toinen nieluista keskenään, mikä aiheutti lineaarimoottorin liikkeen.

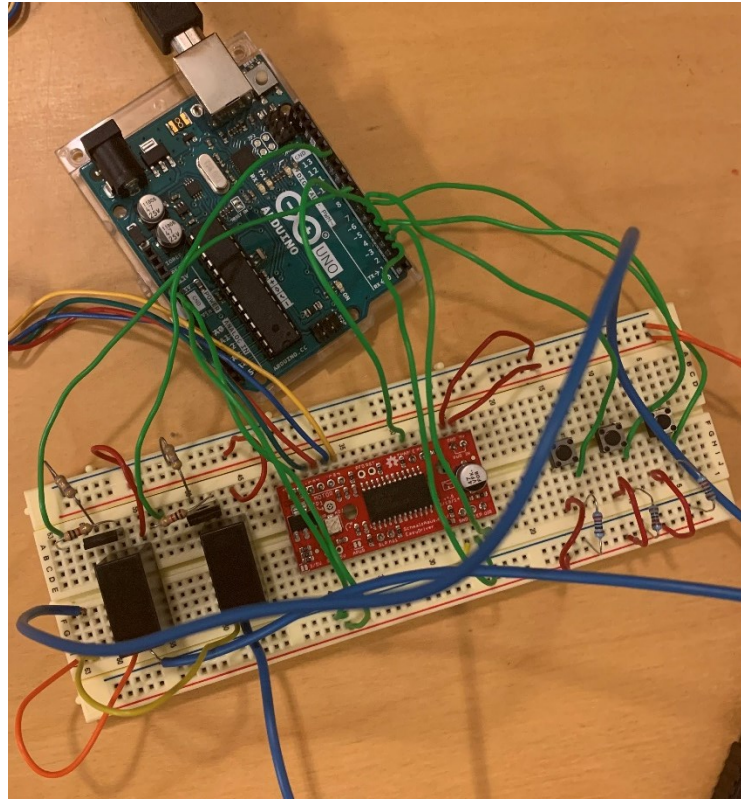
Lineaarimoottorin integroiminen muuhun piiriin aloitettiin suunnittelemalla, miten Arduino voisi ohjata sitä. Koska Arduinon antama ohjaussignaali toimii 5 V:n jännitteellä, päätettiin lineaarimoottorin ohjaus toteuttaa releellä, joka on siis peruseräiteeltään jänniteohjattu kytkin. Valitun releen tulee kestää 38 V:n jännitettä, joten sen ominaisuuksien ja koon takia Arduinon syöttämä 5 V:a ei riittänyt tuottamaan tarpeeksi suurta sähkömagneettista voimaa releen sisäisen kytkimen liikuttamiseen. Releen ohjaamiseen tarvittiin siis 12 V:n jännite, jonka kytkeytyminen releelle toteutettiin BD139-transistorilla. Arduino kytkettiin transistorin kannalle, releen ohjausnavat kytkettiin transistorin kollektorille sarjaan ja transistorin emitteri kytkettiin maapotentiaaliin. Releen ja transistorin kytkentä on esitetty kuvassa 16.



**Kuva 16. Releen ja transistorin kytkentä.**



Kyt kentään lisättiin vastukset RB ja RE varotoimenpiteenä, ettei virta kasva liian suureksi, sillä esimerkiksi ilman vastusta RE virtalähteen syöttämällä virralla on lähes suora reitti maapotentiaaliin. Kuvan 16 mukaisia kytkentöjä rakennettiin kaksi kappaletta, sillä lineaarimoottorin molemmat liikesuunnat vaativat oman ohjauksensa. Lopullinen kytkentä on esitetty kuvassa 17.



**Kuva 17. Lopullinen prototyypikytkentä.**

Kuvassa vasemmalla näkyvät mustina laatikkoina kaksi relettä sekä niiden yläpuolella niitä ohjaavat transistorit. Siniset paksut johtimet kytkeytyvät lineaarimoottorin ohjaimiin.

### 3.2.2 Ohjelmointi

Kierro- ja venytyslaitteessa käytettävä Arduino-kehitysalusta ohjelmoitiin käyttäen Arduinon omaa ohjelmointiympäristöä. Tässä luvussa käsiteltävä ohjelmakoodi on esitetty liitteessä A. Ohjelmointi aloitettiin määrittämällä jokaiselle tarvittavalle sisäänmenolle sekä ulostulolle nimi ja liittimen numero. Nämä numerot vastaavat Arduinon digital input/outputliittimien numeroita. Askelmoottorin ohjaimelle varattiin liittimet 2-4 ja painikkeille varattiin liittimet 5, 6 ja 13. Lineaarimoottorin ohjaukselle varattiin liittimet 11 ja 8. Määrittäminen tehtiin ohjelmakoodilla

```
#define stp 3
#define dir 2
```

```

#define EN 4
#define but_1 5
#define but_2 6
#define but_3 13
#define lin_1 11
#define lin_2 8,

```

jossa *define* on määrittyskomento, *stp*, *dir* ja *EN* ovat moottoriohjaimen sisäänmenoja, *but\_1*, *but\_2* ja *but\_3* ovat painikkeille määritetyt sisäänmenot ja *lin\_1* sekä *lin\_2* ovat lineaarimoottoria ohjaavia ulostuloja.

Tämän jälkeen ohjelmaan asetettiin tarvittavat muuttujat, joista osa toimii tilalippuina. Rajakytkimien tilaa kuvaamaan määritettiin int-muuttujat *limit\_1* sekä *limit\_2*, jotka alustettiin arvoon 0. Start- sekä stop-painikkeen tilaa kuvaamaan määritettiin int-muuttuja *startstop*, joka alustettiin arvoon 0. Laitteen käynnistämistä ja lopetussyklin aloittamista kuvaavat tilaliput asetettiin *int*-muuttujiksi *Start* ja *Stop*, jotka alustettiin arvoon 0. Laitteen pyörimissuunnan määrittämiseksi asetettiin *int*-muuttuja *rotation*, joka alustettiin arvoon *HIGH*.

Määritysten sekä muuttujien alustamisen jälkeen määritettiin jokaisen käytettävän digital input/output -liittimen toiminnallisuus. Tämä määrittäminen tapahtui *void setup()* -funktiossa. Moottoriohjaimen käyttöön varatut liittimet *stp*, *dir* sekä *EN* määritettiin ulostuloiksi (output), painikkeille varatut liittimet *but\_1*, *but\_2* sekä *but\_3* määritettiin sisäänmenoiksi (input) ja lineaarimoottorille varatut liittimet *lin\_1* sekä *lin\_2* määritettiin ulostuloiksi.

Kun kaikki tarvittavat määrittäykset oli tehty, toteutettiin itse ohjelmaa suorittava funktio *void loop()*. Ohjelma toistaa tätä funktiota niin kauan, kunnes jokin tekijä lopettaa ohjelman suorittamisen. Funktion alussa liitin 4 (*EN*) asetetaan arvoon *LOW*, jolloin moottoriohjaimelle menevä enable-signaali saa arvons 0 ja moottoriohjain kykenee ohjaamaan moottoria. Tämän jälkeen painikkeiden arvo luetaan painikkeille varattuihin muuttujiin *limit\_1*, *limit\_2* sekä *startstop* *digitalRead()*-komennon avulla. Muuttujan *startstop* arvoa käytetään hyväksi tilalippujen *Start* ja *Stop* määrittämiseksi. Tilalippujen määrittäminen tehtiin ohjelmakoodilla

```

if ( startstop == HIGH) {
    if ( Start == 0 ){
        Start = 1;
    }
    else {
        Start = 0;
        Stop = 1;
    }
}
}.

```

Painettaessa start/stop-painiketta, muuttuja *startstop* saa tilan HIGH, jolloin if-lauseen ehto toteutuu. Tämän jälkeen ohjelma tarkastelee Start- ja Stop-tilalippuja ja niiden arvoja. Mikäli Start-tilalipun arvo on 0, ohjelma asettaa sen arvoksi 1, joka tarkoittaa kiertoja venytyslaitteen käynnistymistä. Mikäli start/stop-painiketta painaessa Start-tilalippu on arvoltaan 1, asettaa ohjelma Start-lipun arvoon 0 ja Stop-tilalipun arvoon 1.

Askelmoottorin pyörimissuunnan valinta tapahtuu rajakytkimille määriteltyjen muuttujien *limit\_1* ja *limit\_2* avulla. Mikäli jompikumpi edellä mainituista muuttujista saa arvon HIGH, if-lauseen ehto täyttyy ja moottorin pyörimissuunta vaihtuu, eli muuttujan *rotation* arvoa muutetaan arvojen HIGH ja LOW välillä.

Moottoreiden liikuttaminen aloitetaan, kun Start-tilalippu on saanut arvon 1. Lineaari-moottorin liikesuunta määritetään rotation-muuttujan avulla, jolloin sen liikesuunta riippuu askelmoottorin liikesuunnasta. Lineaarimoottori aloittaa pyörimisen, kun liittimen 2 (STP) ulostuloarvoa vaihdellaan arvojen HIGH ja LOW välillä. Moottorin pyörimisnopeus riippuu näiden arvonvaihteluiden nopeudesta. Moottoreiden liikuttaminen on esitetty kuvassa 18.

```
if ( Start == 1) {
  if ( rotation == HIGH ) {
    digitalWrite(lin_1, HIGH);
  }
  else {
    digitalWrite(lin_1, LOW);
  }
  if ( rotation == LOW ) {
    digitalWrite(lin_2, HIGH);
  }
  else {
    digitalWrite(lin_2, LOW);
  }
  digitalWrite(dir, rotation);
  digitalWrite( stp, HIGH );
  delay(1);
  digitalWrite( stp, LOW);
  delay(1);
}
```

**Kuva 18. Moottoreiden liikuttaminen.**

Kuvassa *digitalWrite()*-komennolla asetetaan liittimelle haluttu ulostuloarvo ja *delay*-komennolla asetetaan viive arvon vaihtelun välille.

Mikäli Stop-tilalippu saa arvon 1 kesken ohjelman suorituksen, siirtyy ohjelman suoritus erilliseen funktioon *end\_cycle()*. Funktio on esitetty kuvassa 19.



```

void end_cycle () {
  //Jos ollaan kiertämässä, käännetään moottorin pyörimissuunta, jolloin testikappale palautetaan suoraksi
  if ( rotation == LOW ) {
    rotation = HIGH;
    digitalWrite(lin_1, HIGH);
    digitalWrite(lin_2, LOW);
  }
  //Tarkastellaan perustilan rajakytkintä. Liike pysäytetään, kun raja on saavutettu
  if ( limit_1 == HIGH ) {
    Stop = 0;
    rotation = LOW;
    digitalWrite(lin_1, LOW);
    digitalWrite(lin_2, LOW);
  }
  //Pyöritetään moottoria kunnes raja on saavutettu
  digitalWrite(dir, rotation);
  digitalWrite( stp, HIGH );
  delay(1);
  digitalWrite( stp, LOW);
  delay(1);
}

```

### ***Kuva 19. End\_cycle-funktio.***

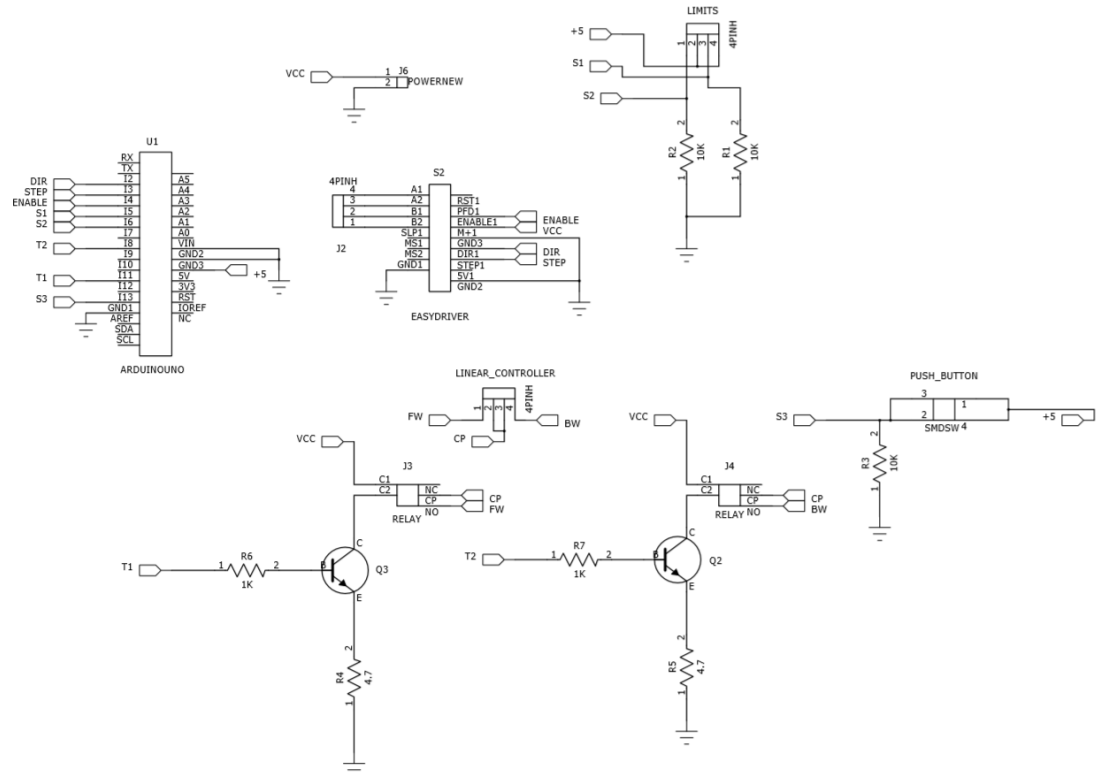
Tämä funktio päättää ohjelman suorittamisen ja tarvittaessa palauttaa testilaitteen takaisin alkutilaansa, eli tilaan, jossa näyte on suorana eikä sitä venytetä. Aluksi funktiossa tarkastetaan mihin suuntaan näytettä ollaan venyttämässä. Mikäli näyte on keskeytyksen hetkellä palaamassa takaisin alkutilaansa, moottoreiden liikuttamista jatketaan, kunnes rajakytkin on saavutettu, jonka jälkeen Stop-tilalippu asetetaan arvoon 0 ja ohjelma lopettaa suorittamisen. Mikäli keskeytyksen hetkellä näyte ei ole palaamassa alkutilaansa, moottoreiden liikesuunta käännetään, jonka jälkeen jäädään odottamaan rajakytkimen saavuttamista.

### **3.2.3 Piirilevyn valmistaminen**

Lopullinen ohjauskytkentä päätettiin rakentaa itsevalmistetulle piirilevylle. Kun kaikki toiminnallisuus oli saatu testattua, aloitettiin piirilevysuunnittelu prototyypikytkennän pohjalta. Suunnittelussa käytettiin hyväksi PADS-ohjelmistoa, joka on piirilevysuunnittelua varten luotu työkalu.

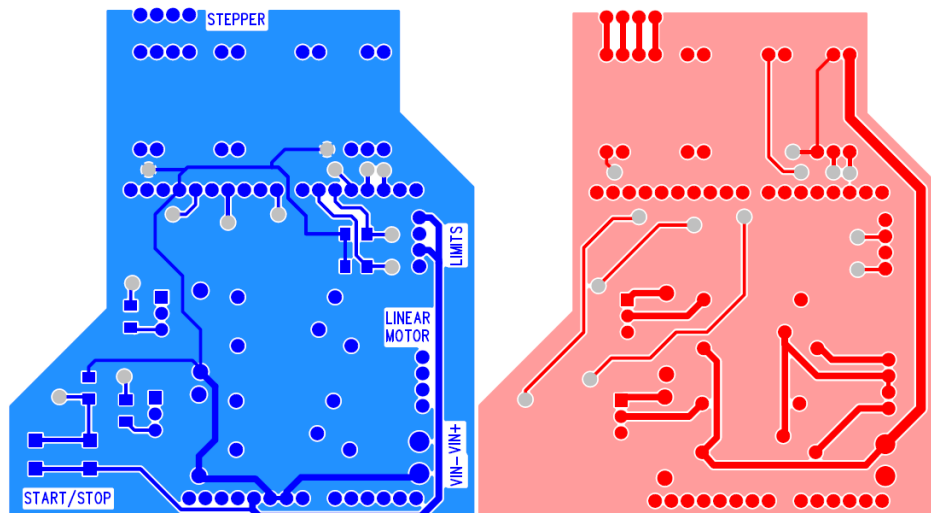
Piirilevystä päätettiin tehdä shield-tyyppinen, eli suoraan Arduinon päälle asetettava. Tähän päädyttiin, sillä Arduinon ja piirilevyn välille tulisi saada 12 liitäntää. Näiden toteuttaminen piikkirimaa hyväksikäyttäen todettiin helpoimmaksi ratkaisuksi. Myöskin piirilevylle asetettavien komponenttien vähäinen määrä mahdollisti sen pienen koon, jolloin shield-tyypin toteuttaminen oli helppoa.

Suunnittelu aloitettiin toteuttamalla prototyypikytkennästä piirikaavio PADS-ohjelmaan. Piirikaavio on esitetty kuvassa 20.



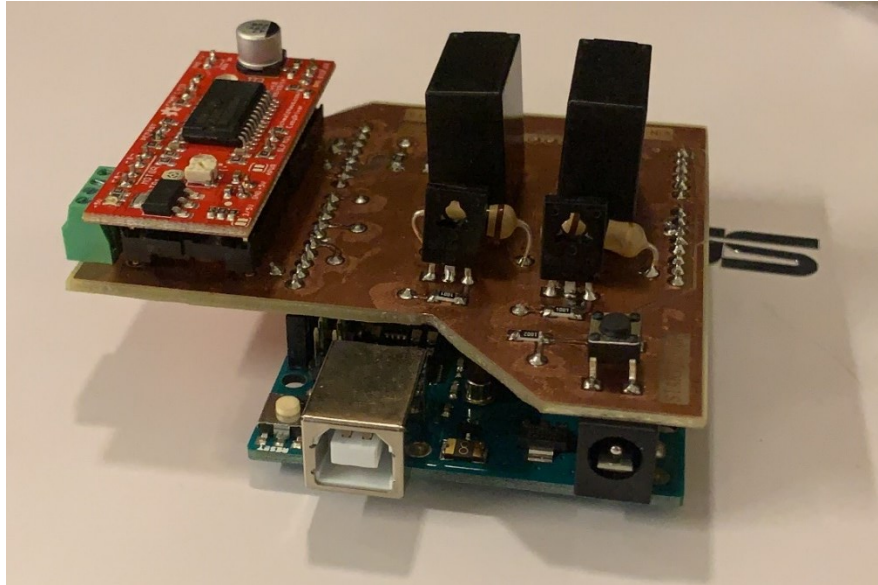
**Kuva 20. Kytkennän piirikaavio.**

Piirikaaviossa on kuvattu koko ohjauspiirin toiminnallisuus. Tämän jälkeen piirikaavion pohjalta suunniteltiin piirilevylayout, joka on esitetty kuvassa 21.



**Kuva 21. Piirilevylayout.**

Piirilevylayoutia hyväksi käyttäen valmistettiin valotuskalvo, jonka avulla valmistettiin piirilevy. Tämän jälkeen piirilevylle juotettiin kiinni tarvittavat komponentit. Valmis piirilevy on esitetty kuvassa 22.

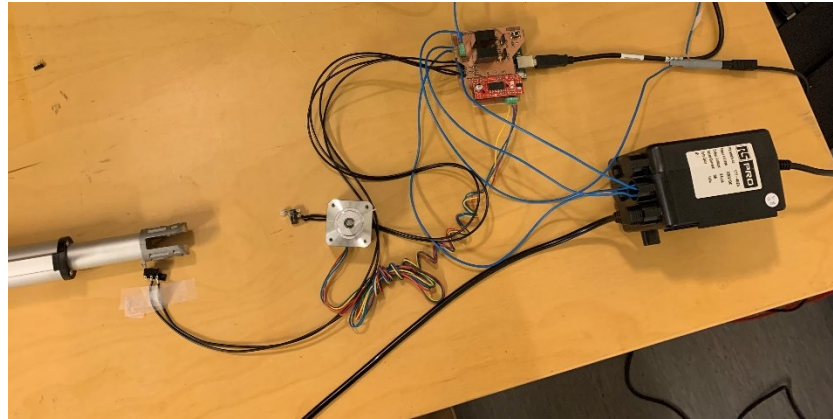


**Kuva 22. Valmis piirilevy.**

Valmis piirilevy asetettiin Arduinon päälle kuvan 22 mukaisesti siten, että siinä olevat piikkirimat uppoavat Arduinon jokaiseen liittimeen. Piirilevyssä käytettiin prototyyppikytkennästä poiketen pintaliitosvastuksia sekä transistorien emitterin ja maapotentiaalin välissä 10 ohmin tehovastuksia. Piirilevyllä toteutettiin vain yksi painike, joka toimii start/stop-painikkeena. Rajakytkimet liitetään piirilevyllä niille varattuihin liittimiin.

### 3.2.4 Laitteen toiminta

Laitteen toimintaa testattiin jo ennen piirilevyn valmistamista, mutta piirilevyn valmistuttua voitiin testata laitteen toimintaa kokonaisuudessaan rajakytkimien kautta, jolloin laite vaihtoi suuntaa itsenäisesti. Testausjärjestelyt on esitetty kuvassa 23.



**Kuva 23. Testausjärjestely.**

Testaus toteutettiin asettamalla rajakytkimet lineaarimoottorin työntövarren etupuolelle ja sivulle kuvan 23 mukaisesti, jolloin työntövarsi aktivoi ne itsenäisesti. Testauksessa laite oli käynnissä useita kymmeniä minuutteja ja se toimi koko testauksen ajan halutulla tavalla.

## 4. JOHTOPÄÄTÖKSET

Tässä kandityössä toteutettiin pelkästään sähköinen osuus kierto- ja venytystestilaitteesta, joten laitteen todellista suorituskykyä ei päästy testaamaan. Laitteen ohjauksen testaaminen kuitenkin osoitti, että laitteella on potentiaalia suoriutua venyvän elektronikan testauksesta, kunhan mekaaninen toteutus ja resistanssia mittaava laitteisto saadaan rakennettua laitteelle. Laitetta testatessa havaittiin, että lineaarimoottorin liikenopeus on huomattavasti odotettua hitaampi, joten testaussyklin kesto kasvaa suunniteltua pidemmäksi. Tämä aiheuttaa ongelmia, sillä yhden testattavan näytteen testaukseen saattaa nyt kulua useita tunteja.

Suurimmat ongelmat työtä tehdessä oli lineaarimoottorin kanssa. Koska datalehdiltä ei löytynyt kaikkia tarvittavia tietoja, täytyi turvautua tavarantoimittajan tekniseen tukeen, jonka vastausajat olivat pitkiä. Tämän lisäksi oikeanlaisen ohjauskytkennän rakentaminen lineaarimoottorille vaati useita tunteja testausta ja suunnittelua.

Koska laite ei ole vielä valmis, täytyy laitetta vielä jatkokehittää huomattavasti. Jatkokehityskohteina ovat muun muassa laitteen nopeuden kasvattaminen sekä mekaaninen suunnittelu ja toteutus, mikä pitää sisällään telineen ja kiinnityskohdat näytteelle sekä askelmoottorin kiinnittämisen lineaarimoottoriin. Myöskin testauslaitteiston rakentaminen on osa jatkokehitystä. Laitteen yhteyteen täytyy siis toteuttaa oma piiri, joka mittaa näytteen resistanssia ja toiminnallisuutta ja taulukoi mitattuja arvoja muistiin. Testauslaitteiston täytyy myös pystyä lähettämään laitteen ohjauspiirille keskeytyssignaali, mikäli resistanssissa tai toiminnallisuudessa havaitaan merkittäviä muutoksia. Tämä keskeytyssignaali aloittaa kuvan 19 mukaisen keskeytyssyklin, joka palauttaa näytteen venytyksen ja kierron alkuperäiseen tilaansa. Testauslaitteistosta täytyy myös saada ulos dataa, joka kertoo, kuinka monta sykliä näytettä on pystytty venyttämään ja kiertämään.

# LÄHTEET

- [1] K. Suganuma, Introduction to Printed Electronics, NY: Springer, New York, 2014, pp. 129.
- [2] K.D. Harris, A.L. Elias, H. Chung, Flexible electronics under strain: a review of mechanical characterization and durability enhancement strategies, *Journal of Materials Science*, Vol. 52, 2016, pp. 2771–2805.
- [3] IPC, IPC-9204 Guideline on Flexibility and Stretchability Testing for Printed Electronics, 2017.
- [4] Firgelli Automations Team, Inner Workings of a Linear Actuator, 2018. Saatavissa (Viitattu 16.10.2019): <https://www.firgelliauto.com/blogs/news/inside-a-linear-actuator-how-a-linear-actuator-works>.
- [5] T.R. Kuphald, Lessons in Electric Circuits Vol. 2 – Alternating Current, 2011, pp. 408-482.
- [6] Arduino. Frequently Asked Questions. Saatavissa (Viitattu 12.10.2019): <https://www.arduino.cc/en/main/FAQ#toc13>.
- [7] RS Pro, Linear actuator 1774514, 1774515, 1774516, 1774517 Datasheet. Saatavissa (Viitattu 12.10.2019): <https://docs-emea.rs-online.com/web-docs/1683/0900766b81683462.pdf>.
- [8] Sparkfun, Easy Driver Hook-up Guide, 2015. Saatavissa (Viitattu: 8.11.2019): [https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/EasyDriver\\_Hookup\\_Guide\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/EasyDriver_Hookup_Guide_Web.pdf).

## LIITE A: OHJELMAKOODI

```

#define stp 3
#define dir 2
#define EN 4
#define but_1 5
#define but_2 6
#define but_3 13
#define lin_1 11
#define lin_2 8

int limit_1 = 0;      //perustilan rajakytkin
int limit_2 = 0;      //venyneen tilan rajakytkin
int startstop = 0;    //Start-stop -painike
int Start = 0;        //Apumuuttuja laitteen käynnistämiseen
int Stop = 0;         //Apumuuttuja lopetussyklin käynnistämiseen
int rotation = HIGH;  //Pyörimissuunta

void setup() {
    //Input/output määrittäminen
    pinMode(stp, OUTPUT);
    pinMode(dir, OUTPUT);
    pinMode(EN, OUTPUT);
    pinMode(but_1, INPUT);
    pinMode(but_2, INPUT);
    pinMode(lin_1, OUTPUT);
    pinMode(lin_2, OUTPUT);
}

void loop() {
    //Enable asetetaan LOW-arvoon, jotta askelmoottori toimii
    digitalWrite(EN, LOW);

    //Painikkeiden arvojen lukeminen
    limit_1 = digitalRead(but_1);
    limit_2 = digitalRead(but_2);
    startstop = digitalRead(but_3);

    //Start-stop -painikkeen tilan tarkastelu
    if ( startstop == HIGH) {

        //Start -painiketta painettu
        if ( Start == 0 ){
            Start = 1;
        }
        //Stop -painiketta painettu
        else {
            Start = 0;
            Stop = 1;
        }
    }

    //Perustilan rajan tarkastelu
    if ( limit_1 == HIGH ) {
        rotation = LOW;
    }

    //Venyneen tilan rajan tarkastelu

```

```

    if ( limit_2 == HIGH ) {
        rotation = HIGH;
    }

    //Moottorin pyöritys
    if ( Start == 1 ) {
        if ( rotation == HIGH ) {
            digitalWrite(lin_1, HIGH);
        }
        else {
            digitalWrite(lin_1, LOW);
        }
        if ( rotation == LOW ) {
            digitalWrite(lin_2, HIGH);
        }
        else {
            digitalWrite(lin_2, LOW);
        }
        digitalWrite(dir, rotation);
        digitalWrite( stp, HIGH );
        delay(1);
        digitalWrite( stp, LOW);
        delay(1);
    }
    //Luetaan apumuuttujaa, joka määrittää, halutaanko moottorin pyöritys lo-
    pettaa
    if ( Stop == 1 ) {
        end_cycle();
    }
}

//Sykliä päätyttyä palautetaan moottorin asento perustilaan
void end_cycle () {

    //Jos ollaan kiertämässä, käännetään moottorin pyörimissuunta, jolloin tes-
    tikappale palautetaan suoraksi
    if ( rotation == LOW ) {
        rotation = HIGH;
        digitalWrite(lin_1, HIGH);
        digitalWrite(lin_2, LOW);
    }

    //Tarkastellaan perustilan rajakytkintä. Liike pysäytetään, kun raja on
    saavutettu
    if ( limit_1 == HIGH ) {
        Stop = 0;
        rotation = LOW;
        digitalWrite(lin_1, LOW);
        digitalWrite(lin_2, LOW);
    }

    //Pyöritetään moottoria kunnes raja on saavutettu
    digitalWrite(dir, rotation);
    digitalWrite( stp, HIGH );
    delay(1);
    digitalWrite( stp, LOW);
    delay(1);
}

```